Name_____

Student ID Number_____930772_____

University of Saskatchewan
Department of Computer Science

CMPT 111.5 – Section 05 – Fraser
Midterm Examination
Nov 2, 2000

Answer all questions in the spaces provided on this exam paper. If you don't have enough space, write on the back of the page - clearly indicate that your answer is continued there. Be sure to pace yourself according to the marks allotted to each question.

This midterm exam is open book.
You may refer to your notes, any computer printouts, and your textbook. Computers and calculators are NOT permitted.

If you don't know the exact answer to a question, write as much you can.
Partial marks will be given.

Good Luck!

Section 1 – True & False / Multiple Choice
Marks 24pts : 2pts each.

T 1. TRUE or FALSE : **char** is a primitive data type in java.

F 2. TRUE or FALSE : Comments in java must start with a double slash - **//**.

F 3. TRUE or FALSE : This statement will generate a compile-time error
```
float f = 16.5;
```

F 4. TRUE or FALSE : Every method in a class must contain a return statement.

F 5. TRUE or FALSE : A StringTokenizer parses a String into smaller Strings called
Tokens. Tokens are always separated by a single space.

F 6. TRUE or FALSE : This loop will repeat 7 times
```
int i = 1;
while (i <= 7) {
     System.out.println(i);
}
```

$\dfrac{5}{6}$

B 7. Which statement best describes the role of the Operating System?
A.)   It is a program that translates java source code into java byte code.
B.)   It is a program that manages system resources, devices, and security.
C.)   It executes program instructions and carries out arithmetic.
D.)   It is where the computer stores things like program instructions and data.

A 8. After the statement : **int num = 6 / 12** is executed, what value is stored in
num?
A.) 0   C.) 12
B.) 6   D.) 0.5

A 9. Which of the following is NOT a valid java identifier?
A.) axialCalc
B.) floats
C.) number
D.) void

is stored in **ans**?

        A.) 30   C.) 11

        B.) 3   D.) 18

B 9. Methods in a user defined class that can be called from any code outside of the class must be declared as :

        A.) private

        B.) public

        C.) protected

        D.) transient

C 10. When a class has overloaded methods, it has :
   A.) more than one method to access the same instance variable.
   B.) more than one method with the same return type.
   C.) more than one method with the same name, but with different parameter lists.
   D.) more than one method with the same parameter list, but with different names.

B & 11. Which of the following is NOT true about objects in java?
   A.) In java new objects are created using a constructor.
   B.) Java is an object oriented, and so all data values are represented using objects.
   C.) Objects can be compared using the relational operators == and !=.
   D.) Objects are instances of a particular class, and are accessed using reference variables.

C 12. You have written a program where you assign the value of an int variable called i to a float variable called f. What is going to happen?
   A.) A compile error when you compile the code.
   B.) A runtime error when you run the code.
   C.) The code will execute without any errors.
   D.) It depends if the value in i is negative or positive.

$\frac{7}{8}$

Section 2 – Short Answer / Code Trace
Marks 45pts : 5pts each.

1. Consider the following piece of code.

```
1 String myS = new String("Just another avalanche");
2
3 String s1 = myS.substring(0,1);
4 String s2 = myS.substring(13,16);
5 String s3 = myS.substring(13);
6
7 myS = s1.concat(s2);
8
9 System.out.println(myS.toUpperCase() + " " + myS.length());
```

What is the value of s1 after line 3 is executed?_____J_____

What is the value of s2 after line 4 is executed?_____ava_____

What is the value of **myS** after line 7 is executed? _____ Java _____ ✓

What will be printed on the screen by line 9? _____ JAVA   4 _____ ✓

$$\frac{5}{5}$$

2. Describe, in one or two sentences, what the following piece of code does.

It sets up a bufferedreader to read from a file and a printstream to write to a file. It will read the line from a specified file, calculate its length and print it's length to a new file. At the end, the total length of all the strings will be displayed on the screen.

What will happen if **in.txt** does not exist?

There will be a run-time error. ✓

What will happen if **out.txt** does not exist?

The computer will create it. ✓

```
BufferedReader br = new BufferedReader(
                new InputStreamReader(
                    new FileInputStream(
                        new File("in.txt"))));

PrintStream ps = new PrintStream(
                new FileOutputStream(
                    new File("out.txt"));

int total = 0;

String s = br.readLine();
while (s != null) {
        total = total + s.length();
        ps.println(s.length());
        s = br.readLine();
}
System.out.println(total);
```

3. Consider the following piece of code :

```
1    int x = 0;
2    x++;
3
```

```java
4    if (x == 1)
5        x = x + 1;
6
7    if (x == 2)
8        x = Math.pow(x,2);
9
10   if (x == 3)
11       x = x--;
12
13   System.out.println("Your Lucky number
                            is " + Math.min(x,6));
```

How does the value of **x** change in line 2? It goes from 0 to 1. ✓

Will line 5 be executed? If yes, what is the new value of **x**? Yes 2 ✓

Will line 8 be executed? If yes, what is the new value of **x**? Yes. 4. ✓

Will line 11 be executed? If yes, what is the new value of **x**? No. ✓

What is printed on the screen in line 13?

~~Your~~ ~~luck~~ Your Lucky number is 4. ✓ $\frac{5}{5}$

4. The following if-else statement calculates whether the int value in **num** is even or odd, and prints an appropriate message on the screen. Rewrite this code so that it uses a switch statement instead of an if-else statement.

```
if ((num % 2) == 0)
    System.out.println(num + " is an even number.");
else
    System.out.println(num + " is an odd number.");
```

```
int i = num % 2;

switch (i) {
    case 0 : System.out.println (num + " is an even number.");
             break;
    case 1 : System.out.println (num + " is an odd number");
             break;
}
```

$\frac{5}{5}$

5. Consider the following piece of code. Assume that a BufferedReader called **kb** has already been declared.

```
float number;
System.out.println(" Enter a number.");
System.out.print(" > ");

try {
```

```
            number = Float.parseFloat(...
            System.out.println("Your number is " + number);

    }
    catch (Exception e) {
            System.out.println("Rainy Day");
    }
```

What is printed on the screen if the user enters a **14.7**? _Your number is 14.7._ ✓

What is printed on the screen if the user enters a **4**? _Your number is 4.0._ ✓

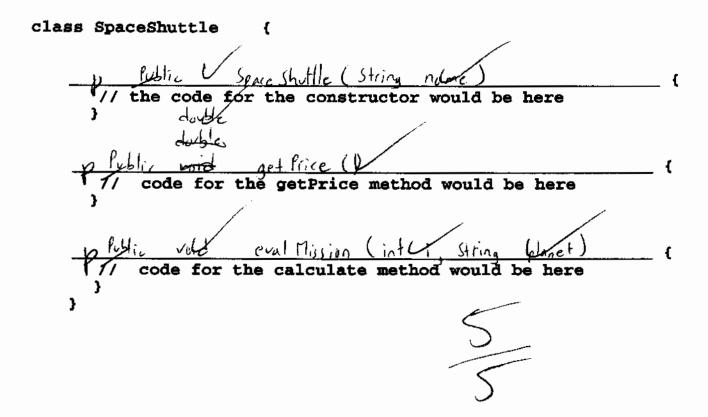What is printed on the screen if the user enters their name? _Rainy Day._ ✓

$\frac{5}{5}$

6. Rewrite the following **while** loop using a **for** loop instead. Show the output that will result when the either loop is executed.

```
int i = -5;
while (i < 0) {
    System.out.println(i + i);
    i++;
}
```

for (int i=-5 ; i<0 ; i++) {
    System. out. println (i + i);
}

Output: -10
         - 8   2/2
         -6
         -4
         -2

3/3

5/5

7. What does the following piece of code do? Do a hand trace, and show your work. For your trace, note how the values of **i**, **t** and **temp** change with each iteration of the loop.

```
String s = new String ("silly string");
String t = new String ("");
int i = 0;
while (i < s.length()) {
    String temp = s.substring(i, i+1);
    t = temp.concat(t);
    i++;
}
System.out.println(t);
```

5/5

	1st	2nd	3rd	4th	5th	6th	7th	8th
String s = silly string								
String t = " "								
int i =	0	1	2	3	4	5	6	7
						" "	s	t

8 Suppose we have a main application class called **Test111**. Inside a method of this class are the following lines of code:

```
SpaceShuttle s = new SpaceShuttle("Discovery");
// Some statements
double d = s.getPrice();
// More statements
s.evalMission(140,"Mars");
```

Given these method calls, you should be able to figure out what three of the method headers inside the **SpaceShuttle** class would look like. Fill in the blanks below. Choose whatever parameter names you like.

```
class SpaceShuttle     {

     public  SpaceShuttle ( String  name )          {
     // the code for the constructor would be here
     }
                double
                double
     public  void  getPrice ()          {
     // code for the getPrice method would be here
     }

     public  void  evalMission (int i, String planet)          {
     // code for the calculate method would be here
     }
}
```

$$\frac{5}{5}$$

9. You have three String variables named **string1**, **string2** and **string3**. Write a single line of code to compute the average length of these three Strings, and output the result to the screen. Partial marks will be given for solutions spanning more than one line..

```
System.out.println ( ( string1.length() + string2.length() + string3.length()) / 3);
```

5

Section 3 – Program Errors
12pts

The following program has exactly twelve errors. Circle each error. Be specific. If you circle entire lines of code you will not get credit. If an error occurs because something is missing, circle the place where the missing thing should go. Circle only twelve errors! If you circle more than twelve, only the first twelve will be marked.

should just be io, don't need other two

```
import java.e_____.*;

class errorProg {

   public static void main(String[] args)
      BufferedReader kb = new BufferedReader(
                          new InputStreamReader(System.in));

      System.out.println("Enter a name for the output file");
      System.out.print(" > ");
      String filename = kb.readLine();

      PrintStream ps = new PrintStream(                    needs new  okay  File
                       new FileOutputStream(filename));
                                                    should be false
      boolean done = true;
      int sum;
                    needs to be initialized
      while (!done) {
         System.out.println("Enter an integer, or Q to quit.");
         System.out.print(" > ");
         String input = kb.readLine();
         input = input.toUpperCase();

         if (input.charAt(0) = "Q")
               done = true;
         else {
               sum = sum + Integer.parseFloat("input");
               PrintStream.println(input);
         }
      } // end while

      System.out.println("The sum of your numbers is + sum);
      System.out.println("You numbers are saved in + filename);

      } // end main                          Should have quotation marks
} // end class                                here
```

1. Write the java code to define a class called **SnowBall**. The class should contain one private instance variable, and three methods. Declare the instance variable to be a float called **size**. Define a constructor method that accepts a float parameter and sets the value of the instance variable. Define a **getSize()** method that returns the **size** of the snowball. Also define a **melt()** method decreases the **size** of the SnowBall by 50%, prints the new size of the snowball on the screen, and returns nothing.

```java
class    Snow Ball {
private  float    size ;

public   Snow Ball ( float  sz ) {
         size = sz ;
}  // end  constructor
public   float   getSize () {
    return  size ;
}  // end  getSize
public  void    melt () {
        size = size / 2.0f ;
        System. out. println ( size );
}  // end  melt
}  // end  class  SnowBall
```